Security Evolution in Vehicular Systems

Dominik Lang, Christopher Corbett, Frank Kargl
Institute of Distributed Systems, Ulm University, Germany
Email: {dominik.lang,christopher.corbett,frank.kargl}@uni-ulm.de

***Abstract*—Modern vehicles contain a complex network of computer systems, which makes security considerations a necessary part of the design process. Due to a vehicle's long deployment phase, static security solutions become obsolete and ineffective over time. Research has mostly focused on how to improve security in vehicles, however, not addressing the need to keep security solutions effective during the entire lifetime of a vehicle. In order to address the changing environment there is a need to create strategies for architectural components, security mechanisms, and update processes that together enable the evolution of the security mechanisms themselves. Our approach is to analyze security mechanisms for how they can fail and what this means from a security evolution perspective. Based on this analysis we can then create solutions in order to evolve deployed security mechanisms over time.**

I. Introduction

Vehicles have evolved into complex computer systems containing up to 100 different Electronic Control Units (ECUs) of which many are interconnected. These ECUs provide functionality for different types of tasks that have different requirements, e. g. low latency or high bandwidth. As a result, the internal network infrastructure in vehicles has evolved to contain several different bus systems that provide different properties for different use cases. In addition, modern automobiles also provide wireless and wired interfaces to the outside world including Bluetooth, WLAN, cellular networking, on-board diagnostic ports (OBDs), USB, and many more. These technologies enable useful services and functionalities, but on the other hand also create new attack surfaces for potential attackers. Especially the wireless interfaces to the outside world open up possibilities for remote attacks.

With these new attack surfaces, it is consensus that security mechanisms are an important aspect to add to the on-board infrastructure, and literature knows a vast number of proposals on how to enhance automotive security (e. g. [1], [2]). However, automotive security is challenged by the very long life cycles of vehicles and also by their safety requirements, which mandate a conservative approach to making changes to deployed vehicles. Thus, for every modification it needs to be ensured that no regressions or flaws are introduced into the safety-critical components, similar to other safety-critical fields such as industrial control systems (ICS) [3]. This conservative approach to automotive IT systems engineering conflicts with the typical approach in IT security, where security mechanisms age, need to be enhanced or replaced, and fast reaction to new attacks is required.

This contradiction is what we address in our work on *security evolution*. Within this work we have split our aims into two separate categories:

1) the identification and categorization of security mechanisms and problems in an automotive systems' life cycle, and
2) the proposal of flexible ways for security mechanisms and architectures to evolve during the life cycle of a vehicle in order to always maintain the required level of security.

The process of security evolution needs to keep security mechanisms in a vehicle updateable and secure at all layers, from hardware to software, and also involves external components, such as public key infrastructures (PKIs). In this paper we present a categorization of domains that security evolution needs to address.

The remainder of this paper is structured as follows: Section II gives a brief overview of security in modern vehicles, Section III categorizes security mechanisms and potential failures, and finally Section IV concludes this paper.

II. Security in Modern Vehicles

Security in vehicles can be categorized into on-board security and V2X communication security (vehicle-to-vehicle and vehicle-to-infrastructure). This work is applicable to both categories, as it focuses on security systems, mechanisms, and components deployed inside a vehicle, which are used to secure the entire infrastructure, i. e. both on-board and V2X security.

ECUs in modern vehicles can be categorized into powertrain, chassis, body and comfort, and driver assistance and safety. As part of the comfort category, a vehicle can contain systems for infotainment (e. g. navigation systems and radio) and telematics units, which are connected to backend servers via a cellular network (e. g. GM's OnStar). Future extensions may introduce other V2X capabilities such as vehicular ad hoc networks (VANETs), for example using dedicated short range radio communication (DSRC) for cooperative safety applications.

The infotainment, telematics, and in general V2X systems provide attack surfaces for remote attacks. Especially telematics units have been targeted by previous work to gain access to the on-board vehicular system via cellular communication [4], [5], [6] and thus allowing the attacker to remain at a safe distance.

On the other hand, the on-board system provides an attack surface for controlling various aspects of the vehicle, including

safety-critical functionalities, such as the brakes and throttle. Attacks have focused on the ability to extract arbitrary data from ECUs by reading their memory and flashing ECUs with malicious code. Attacks have also exploited the ability to replay messages and arbitrarily spoof messages on the CAN bus [6], [7].

As VANETs are not yet deployed, there are no real attacks, but research and development has addressed security (and privacy) from the start. The IEEE 1609.2 standard [8] specifies the use of a PKI in order to equip vehicles with certificates that allow to sign messages between the communicating partners in order to ensure integrity.

In order to protect against malicious modifications of ECU software and unauthorized spoofing of messages, the most important aspects of security for on-board vehicular systems is to provide integrity, authentication, authorization, and availability. Therefore, research has proposed how to secure in-vehicle networks, both in securing the network itself, such as CAN, and in creating a secure architecture for on-board systems [9], [10].

In summary, while risk levels may differ substantially depending on the field of application, security mechanisms ensuring especially integrity, authentication, authorization, and availability (and to a lesser extent confidentiality) require evolutionary capabilities independent of the compartment where they are used.

III. Security Evolution

To approach security evolution in a meaningful way, one first needs to assess and categorize security mechanisms in vehicles and V2X systems. Therefore, we performed a systematic security and risk analysis of security technologies and systems used in vehicles at all layers, e. g. hardware, software, cryptography, architecture, protocols, and network technologies.

Based on this analysis, we examine possible causes that result in a necessary evolution of security mechanisms. We categorize the involved security mechanisms and potential security failures into:

a) ***Configuration****:* One of the biggest problems for security is complexity. Unfortunately security mechanisms often are very complex and difficult to configure. It is easy to make mistakes, for example when configuring firewall and intrusion detection rules, or access control lists. This problem is also evident on the OWASP Top 10 list [11] with ”Security Misconfiguration” being the fifth most critical web application security flaw. One mistake often allows an attacker to completely circumvent the entire security mechanism. Common misconfigurations are: displaying error handling messages back to the user (e. g. SQL errors), enabled directory listings in web servers, running production software in debug mode, using default key material and passwords, or misconfiguring firewall rules.

b) ***Software implementation****:* Software implementations of security mechanisms in ECUs may have design flaws or bugs, which can allow attackers to circumvent a security mechanism. Classical buffer overflows are one example, which were also used in the prominent Heartbleed attack on OpenSSL. Another implementation bug in OpenSSL was used by the FREAK attack, which allowed a man-in-the-middle attacker to enforce the usage of weak RSA keys, which the attacker could then crack.

c) ***Security protocols****:* Security protocols are secure versions of communication protocols, i. e. they protect the interaction between communicating agents; in the case of vehicular systems this applies to both on-board and V2X communication. Communication protocols are subject to various attacks, such as replay, impersonation, and man-in-the-middle attacks. In order to protect against these attacks, security protocols can become very complex, and it is easy to accidentally overlook an aspect or define false assumptions, which then results in possible attacks. These can be subtle mistakes that are only discovered years later. A well-known example is WEP: after it was standardized and deployed, the first attacks were found, which were based on wrong usage of the RC4 cipher [12]. Another well-known example is the Needham-Schroeder protocol [13], where Denning and Sacco pointed out an attack a few years later [14].

d) ***System security****:* System security mechanisms deployed on the ECU level may be insufficient. For example, Address Space Layout Randomization (ASLR) may fail, if not properly implemented or if more sophisticated attacks become available.

e) ***Symmetric cryptography (including hash functions and random number generation)****:* As recently seen with SHA-1 [15], attacks against cryptographic symmetric ciphers and hash functions may become more sophisticated or powerful, and mechanisms considered secure five years ago may not be nowadays. Beyond, even if the algorithm itself is still secure, key lengths may not be appropriate after some some years from now. As an extreme example, many implementations use AES with 128-bit keys; in case of the successful construction of large quantum computers, it is necessary to change the key length to 256-bits due to Grover's algorithm [16], which (from a brute-force search point of view) cuts the number of bits in half, i. e. a 128-bit key can be recovered in 2^{64} steps.

f) ***Asymmetric cryptography****:* The same that applies to symmetric cryptography applies also to asymmetric cryptography. We discuss it separately as the mechanisms are often fundamentally different and key lengths substantially longer. A prominent example in this category is the Logjam attack on TLS [17]: this attack used the fact that many servers were using a single 512-bit group for the Diffie-Hellman key exchange, which the researchers then precomputed. They

were then able to calculate the key from the key exchange. The solution to this problem is to use 2048-bit or larger primes, thus creating the need to update the key material in the field.

*g) **Hardware security modules and functions**:* For performance and cost reasons, some of the mechanisms listed above are cast in (immutable) hardware; in this case replacement is not straightforward. Even if flexible hardware like FPGAs are used, their performance limitations may hinder deployment of more powerful mechanisms, and cost constraints prevent proactive deployment of hardware with spare performance.

*h) **Backend security functions and trusted third parties**:* Security functions embedded into (web-based) backend systems form another part of the security architecture, most notably PKIs, but also authentication and authorization mechanisms. These backend functions are easier to update in case of compromise. However, retaining interoperability with the deployed fleet is one challenge, as well as the failure of a root of trust (e. g. a compromised root CA). The latter case removes the possibility of trustworthy remote interaction with deployed vehicles.

As this list shows, security mechanisms pervade all parts of vehicular architectures. Security evolution capabilities must also become part of all these components in order to be able maintain the state of security. However, the examples given above show that implementing security evolution may not be straightforward in many cases and require further research.

Our next step is to focus on the categories one by one, analyze their requirements, technologies, risk structures, design strategies, methods, and architectures with focus on the evolutionary process of the specific security mechanisms.

Often, security evolution cannot be retrofitted but needs to be an initial capability and part of the system architecture. For example, if modification of keys is required, one needs to identify ways to do this without compromising the integrity of the key store. On the software level, implementations need to be structured in a flexible manner without hard coding assumptions on key size, cipher mode, and the cipher itself.

Security evolution strategies can be split into two different categories: *proactive architectural components* and *secure update mechanisms*. Proactive architectural components are put into place in order to allow the evolution of security mechanisms without compromising their security. They are based on foreseeable security problems and provide proactive means for flexibility, for example by replacing a cipher with a stronger one. On the other hand, secure update mechanisms provide generic strategies on how to perform updates in a deployed system. These strategies entail the requirement for secure and verified updates, and thus especially involve authentication, authorization, and integrity considerations. Considering the compromise of root CAs makes this a non-trivial task.

IV. Conclusion

In this paper we discussed that the modern vehicle is in need of security mechanisms for both on-board and V2X systems. However, the long life cycle of automotive systems contradicts with the ageing of security mechanisms. As a result, deployed security mechanisms cannot guarantee security properties in the long run. Consequently, security solutions need to incorporate solutions on how to evolve deployed security mechanisms in order to keep them up-to-date. This is especially difficult in the domain of safety-critical automotive systems, which require a conservative engineering approach. Moreover, we discussed that the introduction of these security evolution mechanisms are not straightforward and require further research. With this contribution, we highlight the need for security evolution and establish it as a future line of research in automotive security.

References

[1] A. Groll and C. Ruland, "Secure and authentic communication on existing in-vehicle networks," in *2009 IEEE Intelligent Vehicles Symposium*.

[2] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Proceedings of the workshop on embedded security in cars (escar)04*.

[3] F. Kargl, R. van der Heijden, H. Konig, A. Valdes, and M. Dacier, "Insights on the Security and Dependability of Industrial Control Systems," *IEEE Security Privacy*, vol. 12, no. 6, Nov. 2014.

[4] I. Foster, A. Prudhomme, K. Koscher, and S. Savage, "Fast and Vulnerable: A Story of Telematic Failures," 2015.

[5] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," Tech. Rep., Aug. 2011.

[6] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," Tech. Rep., Aug. 2015. [Online]. Available: http://illmatics.com/Remote Car Hacking.pdf

[7] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental Security Analysis of a Modern Automobile," in *2010 IEEE Symposium on Security and Privacy (SP)*, May 2010.

[8] *1609.2-2013 - IEEE Standard for Wireless Access in Vehicular Environments Security Services for Applications and Management Messages*, 2013.

[9] P. Kleberger, N. Nowdehi, and T. Olovsson, "Towards designing secure in-vehicle network architectures using community detection algorithms," in *2014 IEEE Vehicular Networking Conference (VNC)*, Dec. 2014.

[10] M. Wolf, A. Weimerskirch, and T. Wollinger, "State of the Art: Embedding Security in Vehicles," *EURASIP Journal on Embedded Systems*, vol. 2007, no. 1, Jun. 2007.

[11] "OWASP Top 10 - 2013." [Online]. Available: https://www.owasp.org

[12] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: The insecurity of 802.11," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '01. ACM.

[13] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," vol. 21, no. 12.

[14] D. E. Denning and G. M. Sacco, "Timestamps in key distribution protocols," vol. 24, no. 8.

[15] M. Stevens, P. Karpman, and T. Peyrin, "Freestart collision for full SHA-1."

[16] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. ACM.

[17] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thom, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-Bguelin, and P. Zimmermann, "Imperfect forward secrecy: How diffie-hellman fails in practice," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. ACM.